Iterative Timing Recovery and Turbo Equalization

Aravind Nayak, John Barry and Steven McLaughlin

School of ECE, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
Fax: (+1) 404 894 7883
E-mail: {nayak, barry, swm}@ece.gatech.edu

Abstract*:*
Iteratively decodable error correction codes enable operation at very low SNR, which exacerbates the timing recovery problem. We describe a scheme for joint timing recovery and turbo equalization that significantly lowers the SNR requirement when compared to a conventional receiver that separates timing recovery and turbo equalization. We also investigate the Cramér-Rao bound as a tool for evaluating the performance of iterative timing recovery schemes.

Keywords: Timing recovery, turbo equalization, phase-locked loop, low density parity check code, Cramér-Rao bound.

1. Introduction

Conventional timing recovery assumes that the data symbols are uncoded, as it usually operates at high enough SNR so that the instantaneous decisions are reliable enough. The introduction of iterative error correction codes [1] [2] has enabled reliable operation at significantly lower SNR. The iterative decoding technique has been extended to turbo equalization [3], where the equalizer and the decoder cooperate, improving performance further. Consequently, timing recovery now has to operate at an SNR lower than ever before. An example of where this holds is digital magnetic recording, where iterative codes have been proposed to increase the data density.

At low SNR, timing recovery and decoding are intertwined; timing recovery must exploit the structure of the code to get more reliable decisions. Ideally, one would like to jointly perform timing recovery, equalization and decoding. Unfortunately, the complexity would be prohibitive.

Iterative timing recovery presents a viable approximation to the joint problem, where the timing recovery process is embedded inside the turbo equalizer, and the timing recovery process and the turbo equalizer cooperate [4]. Each iteration of the enhanced turbo equalizer is only marginally more complex than that of the conventional turbo equalizer.

We present the system model under consideration in Section 2. Next, we describe conventional PLL-based timing recovery in Section 3. We present the enhanced turbo equalizer which jointly performs timing recovery and turbo equalization in Section 4. Simulation results for the enhanced turbo equalizer are presented in Section 5. We present the Cramér-Rao bound (CRB) for the system model under consideration, and describe the maximum-likelihood timing estimator that achieves the CRB in Section 6. Finally, we conclude in Section 7.

2. System Model

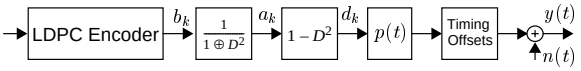

Figure 1: Encoding and partial response channel.

We consider the partial-response system shown in Figure 1 [5], where the readback waveform is

$$y(t) = \sum_k a_k h(t - kT - \tau_k) + n(t), \qquad (1)$$

where T is the bit period, $a_k \in \{\pm 1\}$ are the precoded symbols, $h(t) = p(t) - p(t - 2T)$ is the perfect PR-IV pulse, $p(t) = \sin(\pi t/T)/(\pi t/T)$ is a 0% excess bandwidth pulse, $n(t)$ is additive white Gaussian noise, and τ_k is the unknown timing offset for the k^{th} symbol. We model the timing offset as a frequency offset, according to

$$\tau_k = \tau_0 + k\Delta T. \qquad (2)$$

As shown in Figure 1, message bits are encoded by a serial concatenation of an LDPC encoder and a $1/(1 \oplus D^2)$ precoder.

3. Conventional Timing Recovery

The readback waveform is first filtered by a front-end filter to eliminate out-of-band noise, and the resulting waveform $r(t)$ is sampled according to the output of the timing recovery block to produce samples $\{r_k\}$. Conventional timing recovery is based on a phase-locked loop (PLL). A timing error detector (TED) operates on the previous samples and decisions on these previous samples to estimate the timing error. The output of the TED is accumulated by a PLL, as shown in Figure 2.

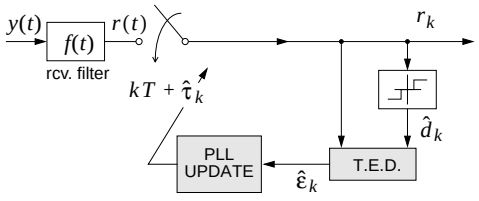

Figure 2: Conventional timing recovery.

A second-order PLL updates according to

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\hat{\epsilon}_k + \beta\sum_{l=0}^{k-1}\hat{\epsilon}_l, \qquad (3)$$

where α and β are the PLL gain parameters, and $\hat{\epsilon}_k$ is the output of the TED which estimates the timing error $\epsilon_k = \tau_k - \hat{\tau}_k$.

The widely used Mueller and Müller TED generates this estimate according to [7]:

$$\hat{\epsilon}_k = \frac{3T}{16}(r_k\hat{d}_{k-1} - r_{k-1}\hat{d}_k), \qquad (4)$$

where $\hat{d}_k$ is an estimate of $d_k = a_k - a_{k-2} \in \{0, \pm 2\}$, typically obtained by a memoryless three-level quantization of r_k. The constant $3T/16$ ensures that there is no bias at high SNR, so that $E[\hat{\epsilon}_k] = \epsilon_k$. Performance can be improved by using soft estimates $\tilde{d}_k$ in place of hard estimates $\hat{d}_k$ in (4) according to

$$\tilde{d}_k = E[d_k|r_k] = \frac{2\sinh(2r_k/\sigma^2)}{\cosh(2r_k/\sigma^2) + e^{2/\sigma^2}}. \qquad (5)$$

To extract the two parameters of interest $\hat{\tau}_0$ and $\hat{\Delta T}$ from the N outputs of the PLL $\{\hat{\tau}_k\}$, we use least squares estimation. Specifically,

$$\begin{aligned}\hat{\Delta T} &= \frac{\langle k\hat{\tau}_k\rangle - \langle k\rangle\langle\hat{\tau}_k\rangle}{\langle k^2\rangle - \langle k\rangle^2},\\ \hat{\tau}_0 &= \langle\hat{\tau}_k - k\hat{\Delta T}\rangle, \qquad (6)\end{aligned}$$

where $\langle x(k)\rangle = \frac{1}{N}\sum_{k=0}^{N-1} x(k)$.

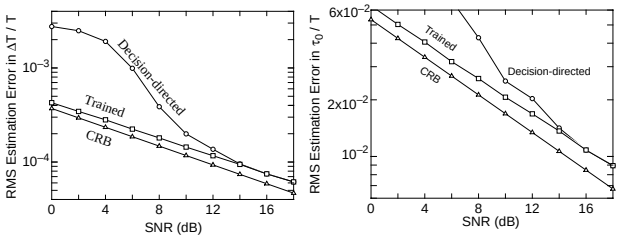

Figure 3: Trained PLL 2 dB away from CRB.

Figure 3 shows the RMS estimation error for the PLL-based system averaged over 10000 blocks of length $N = 250$, $\Delta T/T \sim \text{unif}[0, 0.005]$, $\tau_0/T \sim \text{unif}[0, 0.1]$, α chosen to minimize the RMS estimation error, and $\beta = \alpha^2/4$. Also shown is the Cramér-Rao bound, described in Section 6. The trained system is about 2 dB away from the CRB. For low SNR with no training, we observe a significant performance penalty.

4. Iterative Timing Recovery

A turbo equalizer consists of a soft-in soft-out (SISO) equalizer and a SISO decoder for the LDPC code. The equalizer is based on the BCJR algorithm [9], whereas the LDPC decoder is implemented using the message passing algorithm [2]. A conventional receiver implements the PLL-based timing recovery of Section 3 followed by a turbo equalizer.

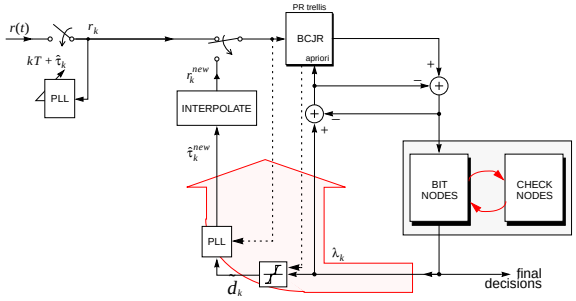

Figure 4: Iterative timing recovery.

We now describe the proposed receiver, which is shown in Figure 4. It begins with a real-time PLL feeding samples $\{r_k\}$ to a turbo equalizer, which feeds soft estimates $\{\tilde{d}_k\}$ to a second PLL which produces improved timing estimates $\{\hat{\tau}_k^{new}\}$. The readback waveform is then effectively resampled at the improved sampling instants using interpolation of the original samples according to

$$r_k^{new} = \sum\nolimits_l r_l p(kT - lT + \hat{\tau}_k^{new} - \hat{\tau}_l). \qquad (7)$$

These new samples are then used in the second iteration of the turbo equalizer. The process then repeats: after each iteration of the turbo equalizer, soft estimates from the turbo equalizer are used to improve the timing estimates, which are then used to interpolate the original samples before going on to the next turbo iteration.

The proposed receiver of Figure 4 is essentially a modified turbo equalizer, with an interpolation step inserted between consecutive iterations. The complexity increase is marginal, because the complexity of interpolation is usually negligible relative to each turbo iteration. It is worth noting that although we perform timing recovery and turbo equalization

jointly, the front-end has remained unchanged, and we still sample the continuous time waveform only once. The modified turbo equalizer is able to correct for poor timing at the front-end PLL.

5. Simulation Results

We consider a rate-8/9 regular (3,27) code of block length 4095, the parity check matrix of which has 3 ones in each column and 27 ones in each row. The channel is precoded PR4, as shown in Figure 1, and we assume AWGN. We assume perfect acquisition, *i.e.*, $\tau_0 = 0$. The frequency offset parameter is $\Delta T = 0.2\%$. A cycle slip is defined to have occured when the timing estimate $\hat{\tau}$ is off from the actual τ by a non-zero integral multiple of T. Within the duration of a block, the τ waveform varies by as much as $8T$, and therefore, cycle slips are quite likely.

Iterative timing recovery corrects cycle slips on its own [4], but convergence can be speeded up significantly by using the following cycle slip detection and correction mechanism. We declare a slip whenever the magnitude of $\zeta_k = \hat{\tau}_k - \hat{\tau}_{k-d}$ exceeds a given threshold H, for some delay d. If a slip is detected, then we use portions of the $\hat{\tau}$ waveform for that iteration not affected by the slip to estimate the frequency offset by the least squares method, and use this estimate to correct the slipped region. After this is done, we revert to memoryless soft-slicer decisions for further timing recovery as opposed to using soft decisions from the turbo equalizer.

If the number of iterations exceeds a certain threshold N_i, we assume the presence of an undetected cycle slip and use the following procedure:

- Construct $\{\zeta_k\}$ where $\zeta_k = \hat{\tau}_k - \hat{\tau}_{k-d}$,
- Compute the mean m_ζ and the standard deviation σ_ζ of the sequence $\{\zeta_k\}$,
- Compute mean m_ζ^p of those ζ_k that lie in the interval $[m_\zeta - \sigma_\zeta, m_\zeta + \sigma_\zeta]$,
- Set $\hat{\Delta T} = m_\zeta^p/d$,
- Use $\hat{\Delta T}$ to resample the whole block.

By doing this, in effect, we exclude the outliers (corresponding to the cycle slip) from the frequency offset computation. This procedure is more computationally intensive than the earlier one.

Figure 5 shows word-error rate vs. SNR for the proposed system with $\alpha = 0.04$, $\beta = \alpha^2/4$, $H = 0.75T$, $d = 100$, with a maximum of 3×10^6 packets being simulated. The threshold N_i was 100 iterations, and after N_i iterations, the more complicated cycle-slip correction algorithm was implemented for at most 25 more iterations. We use the notation x/y to denote the scheduling of iterations, where we have y LDPC iterations before returning to the equalizer block, and we have a total of x such outer iterations, each involving timing recovery followed by turbo equalization. The proposed system gains

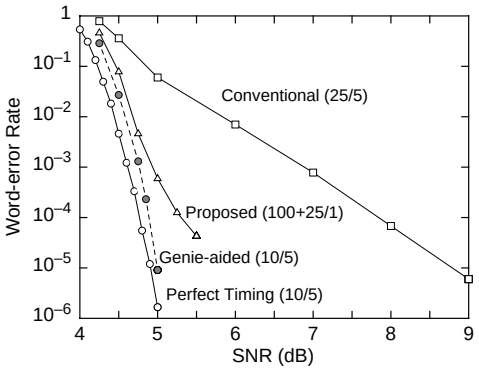

Figure 5: Proposed system 3 dB better.

around 3 dB when compared to the conventional system and is around 1 dB away from the system with perfect timing. Also shown in the same figure is the performance of the genie-aided system, where the PLL has access to the data bits. In effect, this system has a trained timing recovery block followed by a turbo equalizer, and its performance gives a heuristic lower bound to the performance of PLL-based systems. The proposed enhanced turbo equalizer is within 0.5 dB of the genie-aided system.

6. Cramér-Rao Bound and the ML Estimator

For our channel setting and assuming no outer code, the timing recovery problem can be rephrased as follows. Given $\boldsymbol{r} = [r_0\ r_1\ \ldots\ r_{N-1}]^T$, where r_k is the k^{th} uniform sample

$$r_k = \sum a_l h(kT - lT - \tau_l) + n_k, \tag{8}$$

we need to estimate ΔT and τ_0. A natural question that arises is what is the best performance we can expect from any estimator? Cramér-Rao bound answers this question by providing a lower bound on the error variance of unbiased estimators.

In general, the CRB on the error variance of any unbiased estimator $\hat{\boldsymbol{\theta}}(\boldsymbol{x})$ of a parameter $\boldsymbol{\theta}$ based on some observations $\boldsymbol{x}$ is given by

$$E[(\hat{\theta}_i(\boldsymbol{x}) - \theta_i)^2] \geq \boldsymbol{J}_\theta^{-1}(i,i), \tag{9}$$

where $\hat{\theta}_i(\boldsymbol{x})$ and θ_i denote the i^{th} element of $\hat{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\boldsymbol{\theta}$ respectively, the expectation is taken over $\boldsymbol{x}$ and $\boldsymbol{\theta}$, and $\boldsymbol{J}_\theta^{-1}(i,i)$ is the i^{th} diagonal element of the inverse of the Fisher information matrix

$$\boldsymbol{J}_\theta = E\left\{\left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln f_{\boldsymbol{x},\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{\theta})\right]\left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln f_{\boldsymbol{x},\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{\theta})\right]^T\right\}, \tag{10}$$

where the expectation is over $\boldsymbol{x}$ and $\boldsymbol{\theta}$, and $f_{\boldsymbol{x},\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{\theta})$ is the joint probability density of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ [6].

In our case, the parameter to be estimated is $\boldsymbol{\theta} = [\Delta T \ \tau_0]^T$. The Fisher information matrix is

$$\boldsymbol{J}_\theta = \frac{1}{\sigma^2 T^2}\left(\frac{2\pi^2}{3}-1\right)\begin{bmatrix} \frac{(N-1)N(2N-1)}{6} & \frac{N(N-1)}{2} \\ \frac{N(N-1)}{2} & N \end{bmatrix}, \tag{11}$$

where σ^2 is the variance of the noise $\{n_k\}$.

Therefore, the CRB is given by

$$\begin{aligned} \frac{E[(\Delta T - \hat{\Delta T})^2]}{T^2} &\geq \frac{12\sigma^2}{\left(\frac{2\pi^2}{3}-1\right)(N-1)N(N+1)} \\ \frac{E[(\tau_0 - \hat{\tau_0})^2]}{T^2} &\geq \frac{2\sigma^2(2N-1)}{\left(\frac{2\pi^2}{3}-1\right)N(N+1)}. \end{aligned} \tag{12}$$

The CRB can be achieved in the trained case by an ML estimator that chooses $\hat{\tau}_0$ and $\hat{\Delta T}$ to minimize

$$J(\hat{\tau}_0, \hat{\Delta T}; \boldsymbol{a}) = \int_{-\infty}^{\infty} (r(t) - \sum a_k p(t - kT - k\hat{\Delta T} - \hat{\tau}_0))^2 \, \mathrm{d}t, \tag{13}$$

where $\boldsymbol{a} = [a_0 \ a_1 \ \ldots \ a_{N-1}]^T$. This minimization

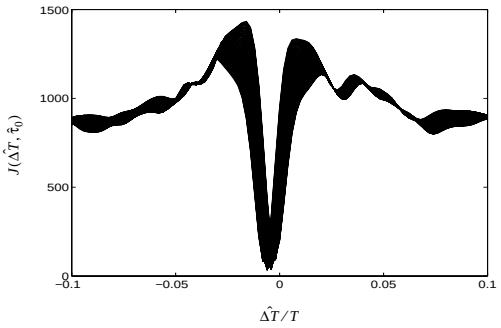

Figure 6: Gradient descent not suitable.

can be implemented by gradient descent. The cost function is plotted as a function of $\hat{\Delta T}$ in Figure 6 for values of $\hat{\tau}$ from $-0.5T$ to $0.5T$. Along the other direction, *i.e.*, along the $\hat{\tau}_0$ axis, the cost function is parabolic in nature. It is evident that gradient descent is not suited for minimization along $\hat{\Delta T}$ because of its sensitivity to initialization, and also because the minimum occurs in a narrow valley.

In the trained case, these issues can be addressed by using the Levenberg-Marquardt (LM) method, which is a combination of gradient descent and Newton's method [8]. As opposed to gradient descent, the LM method moves farther in directions where the magnitude of the gradient is lesser, thus reducing rattling in the long, narrow valley. The LM iterations are initialized using the estimates $\hat{\Delta T}$ and $\hat{\tau}_0$ as described in Section 3. Let

$$\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{a}; \boldsymbol{\theta}) + \boldsymbol{n}, \tag{14}$$

where $\boldsymbol{f}$ is some function that satisfies Equation 8 and $\boldsymbol{n} = [n_0 \ n_1 \ \ldots \ n_{N-1}]^T$. Denote the k^{th} element of $\boldsymbol{f}$ by f_k. At the i^{th} iteration, let $\boldsymbol{\theta}_i$ be the current estimate. To get the next estimate $\boldsymbol{\theta}_{i+1}$,

- compute $\boldsymbol{d} = \sum_k (f_k(\boldsymbol{a}; \boldsymbol{\theta}_i) - \hat{r}_k)[\nabla \boldsymbol{f}(\boldsymbol{a}; \boldsymbol{\theta}_i)]_k$,
- compute $\boldsymbol{H} = \sum_k [\nabla \boldsymbol{f}(\boldsymbol{a}; \boldsymbol{\theta}_i)]_k [\nabla \boldsymbol{f}(\boldsymbol{a}; \boldsymbol{\theta}_i)]_k^T$,
- update $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - (\boldsymbol{H} + \lambda \mathrm{diag}[\boldsymbol{H}])^{-1}\boldsymbol{d}$,
- compute $E(\boldsymbol{\theta}_{i+1}) = \sum_k (f_k(\boldsymbol{a}; \boldsymbol{\theta}_{i+1}) - \hat{r}_k)^2$,
- update λ: if E, the error, has increased, retract the step and increase λ by a significant factor; if E has decreased, accept the step and decrease λ by the same factor.

Here, $[\nabla \boldsymbol{x}]_k$ denotes the gradient of the k^{th} element of $\boldsymbol{x}$. λ controls the relative weight we give to gradient descent and to Newton's method. More weight is given to the Newton's method when we are in the right direction, and to gradient descent otherwise.

7. Conclusion

We proposed an iterative timing recovery scheme for LDPC-encoded PR channels that jointly performs timing recovery and turbo equalization by embedding the timing recovery block inside the turbo equalizer. Simulation results show that iterative timing recovery significantly outperforms a conventional receiver that separates timing recovery and turbo equalization. The enhanced turbo equalizer performs to within 0.5 dB of a genie-aided conventional receiver where the timing recovery block has access to the transmitted data. We also presented the Cramér-Rao bound as a means for evaluating the efficacy of timing recovery schemes, and described the trained ML estimator that achieves the CRB.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes," *Proc. of the IEEE Intern. Conf. on Commun. 1993*, vol. 2, pp. 1064-1070, May 1993.

[2] R. Gallager, "Low-Density Parity-Check Codes," *IRE Trans. on Info. Theory*, vol. IT-8, pp. 21-28, Jan 1962.

[3] D. Raphaeli and Y. Zarai, "Combined Turbo Equalization and Turbo Decoding," *Proc. of the IEEE Global Telecommun. Conf. 1997*, vol. 2, 639-643, Nov 1997.

[4] A. Nayak, J. Barry and S. McLaughlin, "Joint Timing Recovery and Turbo Equalization for Coded Partial Response Channels," *IEEE Trans. Magnetics*, vol. 38, no. 5, pp. 2295-2297, Sept 2002.

[5] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML System for Digital Magnetic Recording," *IEEE J. Sel. Areas in Commun.*, vol. 10, no. 1, pp. 38-56, Jan 1992.

[6] H. Van Trees, *Detection, Estimation, and Modulation Theory*, vol. 1, pp. 72-85, John Wiley and sons, 1968.

[7] K. Mueller and M. Müller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Trans. Commun.*, vol. com-24, no. 5, pp. 516-31, May 1976.

[8] S. Roweis, "Levenberg-Marquardt Optimization," at http://www.cs.toronto.edu/∼roweis/notes/lm.pdf.

[9] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. of Info. Theory*, vol. IT-20, pp. 284-287, Mar 1974.